IJCTA, 9(12), 2016, pp. 5673-5680

Unified Approach to Enhance Security Over *e*-Business Transactions

***R. Rajalakshmi *B. Charanya**

Abstract : In this digital world, *e*-business has gained more popularity, but it is associated with many security issues. The various risks related to online business include transaction risks, data storage risks, privacy risks and transmission risks. In this paper, an unified approach has been proposed to provide enhanced security over e-business transactions with a detailed performance analysis of various cryptographic algorithms. The performance of the proposed approach is studied by conducting various experiments on a real world problem, and an optimal algorithm has been suggested that has minimum transmission risk.

Keywords : Transmission security, Message Digest, *e*-business, XML messages, Algorithm Analysis.

1. INTRODUCTION

The Electronic Business, also known as "*e*-Business" is defined as the utilization of Information and Communication Technologies (ICT) in support of all the activities of business. These methods enable companies to link their internal and external data processing systems more efficiently and flexibly, to work more closely with suppliers and partners, and to better satisfy the needs and expectations of their customers. E- business involves many business processes such as electronic purchasing, processing orders electronically, handling customer service, and cooperating with business partners. In implementing *e*-business solutions, there are various benefits such as quick and easy communication, 24 x 7 operation support, efficient methods for processing and payment there by reducing the cost etc. At the same time, the e-business involves many security risks as the business activities are carried out over the Internet. Eavesdroppers always look for data traveling over wires and hence, there is a need for security techniques at different levels. In this paper, we have proposed a robust system from a detailed performance analysis of cryptographic algorithms and suggested the optimal one by conducting various experiments by considering a real world application like tender notice and purchase order processing.

Transmission security is the capability to send a message electronically from one computer system to another computer system so that only the intended recipient receives and reads the message and the message received is identical to the message sent. Unfortunately, transmission interceptions are inevitable. Even though, we can minimize the risks of transmission interception, we can never, under any circumstances, completely rule it out. However, we can build a system to deter people from attempting to break in, and make it costly for the hacker to enter.

The three main security issues related to any online business include the following :

1. Verifying the identity of the person you are doing business with.
2. Ensuring that messages you send and receive have not been tampered with.
3. Obtaining evidence of the date, time and place at which a contract was made.

These issues are addressed by encryption methods, digital signature and with certification authorities. There are several types of attacks on *e*-business transactions. Some attacks aim at gaining specific information on individuals or companies, where as some other type of attacks try to shut down the network to make it unavailable thereby causing business to loss on revenue. The types of attacks include malicious code, password stealing, instant messaging replay, denial of service attacks etc.

* VIT University, Chennai (mail: rajalaxmi.s@gmail.com and (bcy18@gmail.com)

In practice, there are several types of file transmissions most users of *e*-business perform, including the transmission of files through FTP (file transfer protocol), submitting forms by a Web server, and sending e-mail. Information transferred in this way should be encrypted before transmission. Transferring unencrypted files with these methods means the files travel as plain text, ready to be intercepted and interpreted by anyone. Clearly, encrypting files for transmission adds a level of inconvenience, but to secure the transmission, this inconvenience is unavoidable. Unfortunately, security decisions always involve a trade-off between security and convenience. In this paper, we have proposed a method to make secure file transmission between the sender and the receiver using cryptographic algorithms, which we choose after an extensive analysis of the performance of the existing algorithms. The optimal ones are chosen for our application of *e*-business transactions.

The paper is organized as follows: Section II discusses the related work in this field followed by the proposed methodology in Section III. The implementation details are discussed in Section IV with detailed analysis followed by conclusion in Section V.

2. RELATED WORKS

In the recent years, there is an exponential increase in the number of internet users. This change has made many organizations to extend their services over web and to provide facilities for making e-transactions. However, there is no unique security mechanism for these electronic transactions and they are subjected to various risks. In order to address the challenges involved in e-transaction, various approaches were followed by many researchers and this section discusses some of the related works.

The credentials for securing the transactions are generally stored in local computers, which may lead to misuse. To avoid this, Hakim Fourar-Laidi [2], proposed a framework in which, usage of smart cards have been suggested to store the credentials. In that model, the keys are stored in the smart card, thereby allowing the users to perform authentication, integrity checking using those keys in a secured way. The disadvantage of this method is that, it needs a separate card to be maintained by the user to perform transactions. To improve and to adopt to new technologies such as e-commerce and e- government in Saudi Arabia, Said S. Al-Gahtani [3], made an attempt to study the Technology Acceptance Model (TAM). In his approach, he integrated three constructs viz., trust, credibility and risk. This model has validated with the available settings in Saudi Arabia. As the number of online banking users keep on increasing, the financial institutions are looking for security solutions for their transactions. Saad M. Darwish and Ahmed M. Hassan [4], have proposed a model using Identity Based mediated RSA (IB-mRSA) technique combined with one-time password for authenticating the clients of on-line banking transactions. This approach requires more number of rounds for computation, and not suitable situations in which many attackers are involved.

As many e-commerce business models do not provide fairness to customers for several types of digital items, a fairness solution was proposed by Piva et. al [6]. To test the statistical significance of relationship between the perceptions of customers about the internet banking, a framework has been developed by Haque [7], considering the facts about the Malaysian banking customers and e-banking solutions at Malaysia. To support e-commerce transactions, a security middleware solution has been developed, in which the main fair exchange and nonrepudiation security services has been provided. The middelware solution, Fair Integrated Data Exchange Services (FIDES) was proposed by Nenadic [8] by including authentication and confidentiality components, but not implemented in real time. In e-transactions, various attacks such as data base exploits, sniffing attacks are possible. A study has been done by Prandya [9] addressing the issues. Accountability and acknowledgement are necessary for B2B interactions and the mechanism should ensure fairness among the participating parties. To support fair non-repudiable B2B interactions based on a trusted delivery agent, a flexible framework has been proposed by Robinson [10], in which the role of delivery agent can be changed dynamically depending on the capabilities of end user requirements.

In this paper, a detailed study is conducted with various cryptographic algorithms to securely transmit the messages over internet and time complexity and computational complexity are analyzed for each algorithm.

3. BACKGROUND

In this section, we present the existing algorithms and their applications.

A. Message Digest

A message digest is a compact digital signature for an arbitrarily long stream of binary data. An ideal message digest algorithm would never generate the same signature for two different sets of input, but achieving such theoretical perfection would require a message digest as long as the input file.

MD2, MD4 and MD5 are some of the standard message digests used for creating digital signatures. They are meant for digital signature applications where a large message has to be "compressed" in a secure manner before being signed with the private key. All three algorithms take a message of arbitrary length and produce a 128-bit message digest. [5]

1. **SHA :** The SHA (Secure Hash Algorithm) hash functions refer to algorithms for computing a condensed digital representation (known as a message digest) that is, to a high degree of probability, unique for a given input data sequence (the message). These algorithms are called "secure" because for a given algorithm, it is computationally infeasible to find a message that corresponds to a given message digest and also difficult to find two different messages that produce the same message digest. Any change to a message will, with a very high probability, results in a different message digest.
2. **SHA-1 :** SA-1 (as well as SHA-0) produces a 160-bit digest from a message with a maximum length of 264 " 1 bits and is based on principles similar to those used by Ronald L. Rivest of MIT in the design of the MD4 and MD5 message digest algorithms.
3. **SHA-2 :** The five algorithms denoted SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512, are cryptographic hash functions designed by the National Security Agency (NSA). The latter four variants are sometimes collectively referred to as SHA-2.

   SHA-256 and SHA-512 are novel hash functions computed with 32- and 64-bit words, respectively. They use different shift amounts and additive constants, but their structures are otherwise virtually identical, differing only in the number of rounds. SHA-224 and SHA-384 are simply truncated versions of the first two, computed with different initial values.

   Applications of Secure Hash Algorithms : SHA-1, SHA-224, SHA-256, SHA-384, and SHA- 512 are the secure hash algorithms required by law for use in certain Government applications, including use within other cryptographic algorithms and protocols, for the protection of sensitive unclassified information. SHA-1 is also used by private and commercial organizations.
4. **MD2 :** Message Digest Algorithm 2 (MD2) is a cryptographic hash function developed by Ronald Rivest. The 128-bit hash value of any message is formed by padding it to a multiple of the block length on the computer (128 bits or 16 bytes) and adding a 16-byte checksum to it. For the actual calculation, a 48-byte auxiliary block and a 256- byte table generated indirectly from the digits of the fractional part of pi are used. Once all of the blocks of the (lengthened) message have been processed, the first partial block of the auxiliary block becomes the hash value of the message.
5. **MD5 :** In cryptography, MD5 (Message-Digest algorithm 5) is a widely used cryptographic hash function with a 128-bit hash value. MD5 has been employed in a wide variety of security applications, and is also commonly used to check the integrity of files. An MD5 hash is typically a 32- character hexadecimal number. The size of the hash—128 bits—is small enough to contemplate a birthday attack [4]. Because MD5 makes only one pass over the data, if two prefixes with the same hash can be constructed, a common suffix can be added to both to make the collision more reasonable.

   MD5 processes a variable-length message into a fixed- length output of 128 bits. The input message is broken up into chunks of 512-bit blocks; the message is padded so that its length is divisible by 512. The

padding works as follows: first a single bit, 1, is appended to the end of the message. This is followed by as many zeros as are required to bring the length of the message up to 64 bits less than a multiple of 512. The remaining bits are filled up with a 64-bit integer representing the length of the original message.

Applications of message digest : MD5 digests have been widely used in the software world to provide some assurance that a transferred file has arrived intact. For example, file servers often provide a pre- computed MD5 checksum for the files, so that a user can compare the checksum of the downloaded file to it.

MD5 is widely used to store passwords. A number of MD5 reverse lookup databases exist, which make it easy to decrypt password hashed with plain MD5. Also, it is a good idea to apply the hashing function (MD5 in this case) more than once increases the time needed to encode a password and discourages dictionary attacks.

B. Digital Signature

In cryptography, a digital signature or digital signature scheme is a type of asymmetric cryptography used to simulate the security properties of a signature in digital, rather than written form. Digital signature schemes normally give two algorithms, one for signing which involves the user's secret or private key, and one for verifying signatures, which involves the user's public key. The output of the signature process is called the "digital signature."

Digital signatures, like written signatures, are used to provide authentication of the associated input, usually called a "message." Messages may be anything, from electronic mail to a contract, or even a message sent in a more complicated cryptographic protocol. Digital signatures are used to create public key infrastructure (PKI) schemes in which a user's public key (whether for public-key encryption, digital signatures, or any other purpose) is tied to a user by a digital identity certificate issued by a certificate authority. PKI schemes attempt to unbreakably bind user information (name, address, phone number, etc.) to a public key, so that public keys can be used as a form of identification.

Digital Signature Algorithm

The Digital Signature Algorithm (DSA) is a United States Federal Government standard or FIPS for digital signatures. It was proposed by the National Institute of Standards and Technology (NIST) in August 1991 for use in their Digital Signature Standard (DSS), specified in FIPS 186, adopted in 1993. The DSA algorithm consists of a private key that only the originator of the document (signer) knows and a public key.

DSA is a public key algorithm; the secret key operates on the message hash generated by SHA-1; to verify a signature, one recomputes the hash of the message, uses the public key to decrypt the signature and then compare the results. The key size is variable from 512 to 1024 bits which is adequate for current computing capabilities as long as you use more than 768 bits.

C. Encryption Techniques

A key is a piece of information that is used to convert plain text into secret text. These are broadly classified into private and public key encryption algorithms. One veryimportant feature of a good encryption scheme is the ability to specify a 'key' or 'password' of some kind, and have the encryption method alter itself such that each 'key' or 'password' produces a different encrypted output, which requires a unique 'key' or 'password' to decrypt. This can either be a 'symmetric' key (both encrypt and decrypt use the same key) or 'asymmetric' (encrypt and decrypt keys are different).

Encryption mainly deals with converting a plain text into a "cipher text", which at the first glance looks meaningless. This can be decrypted only when the keys are known.

Symmetric Key based Encryption –Triple DES

In cryptography, Triple DES is a block cipher formed from the Data Encryption Standard (DES) cipher by using it three times.

Algorithm

When it was found that a 56-bit key of DES is not enough to guard against brute force attacks, TDES was chosen as a simple way to enlarge the key space without a need to switch to a new algorithm. The use of three steps is essential to prevent meet-in-the-middle attacks that are effective against double DES encryption.

The simplest variant of TDES operates as follows :

DES(k3;DES(k2;DES(k1;M))), where M is the message block to be encrypted and k1, k2, and k3 are DES keys. This variant is commonly known as EEE because all three DES operations are encryptions. In order to simplify interoperability between DES and TDES the middle step is usually replaced with decryption (EDE mode): DES(*k*3 ; DES " 1(*k*2; DES(*k*1 ; M))) and so a single DES encryption with key *k* can be represented as TDES-EDE with *k*1 = *k*2 = *k*3 = *k*. The choice of decryption for the middle step does not affect the security of the algorithm.

Security of TDES

In general TDES with three different keys (TDES) has a key length of 168 bits: three 56-bit DES keys (with parity bits TDES has the total storage length of 192 bits), but due to the meet-in-the-middle attack the effective security it provides is only 112 bits. A variant, called two-key TDES (2TDES), uses *k*1 = *k*3, thus reducing the key size to 112 bits and the storage length to 128 bits. However, this mode is susceptible to certain chosen-plaintext or known-plaintext attacks and thus it is officially designated to have only 80- bits of security.

Asymmetric Key based Encryption –RSA

RSA algorithm is a public-key cryptosystem defined by Rivest, Shamir, and Adleman. In the case of the RSA encryption algorithm, it uses very large prime numbers to generate the public key and the private key. Although it would be possible to factor out the public key to get the private key (a trivial matter once the 2 prime factors are known), the numbers are so large as to make it very impractical to do so. Thus RSA is very secure.

Security of RSA

The security of the RSA cryptosystem is based on two mathematical problems: the problem of factoring large numbers and the RSA problem. Full decryption of an RSA cipher text is thought to be infeasible on the assumption that both of these problems are hard, *i.e*., no efficient algorithm exists for solving them. Providing security against partial decryption may require the addition of a secure padding scheme.

The RSA problem is defined as the task of taking eth roots modulo a composite n: recovering a value m such that *c* = me mod *n*, where (*e*, *n*) is an RSA public key and *c* is an RSA ciphertext. Currently the most promising approach to solving the RSA problem is to factor the modulus *n*. With the ability to recover prime factors, an attacker can compute the secret exponent d from a public key (*e*, *n*), then decrypt c using the standard procedure. To accomplish this, an attacker factors *n* into *p* and *q*, and computes (*p*-1) (*q*-1) which allows the determination of *d* from *e*.

No polynomial-time method for factoring large integers on a classical computer has yet been found, but it has not been proven that none exists.

RSA keys are typically 1024–2048 bits long. Therefore, it is generally presumed that RSA is secure if n is sufficiently large.

4. PROPOSED APPROACH TO ENHANCE SECURITY OF E-TRANSACTIONS

In any *e*-business transactions, securing data transmitted over internet is mandatory. The proposed system is designed to provide enhanced security for *e*-business transactions by making a detailed performance analysis on various aspects of cryptographic algorithms and choosing the optimal one for authentication and confidentiality.

A. Secure File Transmission

In practice, there are several types of file transmissions most users perform, including the transmission of files through FTP (file transfer protocol), submitting forms by a Web server, and sending e-mail.

Information transferred in this way should be encrypted before transmission. Transferring unencrypted files with these methods means the files travel as plain text, ready to be intercepted and interpreted by anyone. Clearly, encrypting files for transmission adds a level of inconvenience, but to secure the transmission, this inconvenience is unavoidable. Unfortunately, security decisions always involve a trade-off between security and convenience. The two major aspects in providing security include authentication and confidentiality. After extensively researching the performance of cryptographic algorithms and considering the security issuesarising in file transfer across the network, we have come up with a unified approach for secure transmission of data for e-business transactions.

B. Unified Approach for providing Enhanced Security

The proposed system is designed to provide enhanced security for e-business transactions by making a performance analysis of the existing set of cryptographic algorithms and choosing the optimal ones for authentication and confidentiality, which are the two major aspects in achieving security. In computer security, authentication is the process of attempting to verify the digital identity of the sender of a communication such as a request to log in with a password. The weakness in this system for transactions that are significant (such as the exchange of money) is that passwords can often be stolen, accidentally revealed, or forgotten. For this reason, Internet business and many other transactions require a more stringent authentication process. The use of digital certificates issued and verified by a Certificate Authority (CA) as part of a public key infrastructure is considered likely to become the standard way to perform authentication on the Internet. Confidentiality is defined as "ensuring that information is accessible only to those authorized to have access" and is one of the cornerstones of Information security.

To provide enhanced security to the *e*-transactions, the following steps are done in the proposed system. Initially, various cryptographic algorithms' performance analysis has been studied. Next, the user information which is submitted as a HTML form data is used to create an XML file. XML is a W3C-recommended general-purpose markup language that supports a wide variety of applications. The purpose of XML is to create a skeletal construct for exchanging complete, often complex, types of data reliably. Instead of sending the data directly, these XML (Extensible Markup Language) files are transmitted in a secured way. For providing authentication, among the several techniques, the optimum technique is chosen and message digest is created for this XML file. Then digital signature is created by generating key pairs. After this process, session keys are also generated and by using optimal and appropriate cryptographic algorithm, the XML file is encrypted using the session key. Finally, the encrypted session and the file are transmitted over the internet. At the receiver end, the digital signature is verified and the decryption is performed to obtain the key and data.

The working of the proposed system is detailed below. In our system for securing e-business transactions, we first give the user options to select the type of transaction. Here, we have two input forms Tender Notice and Purchase Order. In Tender Notice, we obtain user information like Company Name, Supplier Name, Phone number, Customer ID, Item Required, Quantity, Approximate cost, Turn over, Place and Date. In Purchase Order, the details like Customer Number, Code, Order, Description, Date and Amount are obtained. As these obtained details contain sensitive information, they have transmitted securely. Instead of operating on individual fields of this form, XML file usage has been proposed, on which cryptographic techniques have been applied to increase the security. Once these details are obtained using HTML forms, we generate an XML file by hierarchically arranging the form data. For the creation of this XML file, we use the Java Document Object Model.

Once the XML file is created, a message digest is generated using the SHA-1 algorithm, which is chosen from a performance analysis of various cryptographic algorithms. (Before selecting this particular algorithm, all the other algorithms have been applied and the performance has been studied for each algorithm in terms of computational complexity and time complexity). This message digest authenticates the sender of the data. A digital signature is then created for the message digest using DSA. The digital signature is like an electronic signature which identifies the sender uniquely. Next, the authenticated sender data is encrypted using the 3DES algorithm. Triple DES encryption has three times the complexity of DES and is hence a much more secure option for our application. The session key used for this encryption is dynamically obtained from the sender each time the data has to be transmitted.

To make the transmission more secure and to add robustness to our system, we perform encryption of the session key itself using RSA algorithm. The RSA has a prime public-private key pair which is generated for every encryption. Also it uses a one-way mathematical function which makes it very difficult for the intruders to decipher the keys and in turn decrypt the data.

Once the encryption of the key and data is performed, we securely transmit the same over the network. At the receiving end, on verifying the sender's digital signature, the receiver is asked for the session key. Only when the sender and receiver keys match, is the data decrypted. On entering the appropriate keys, the receiver selects the file to be decrypted. After this, the decrypted contents can be obtained at the receiver end. By this way, an enhanced security mechanism is provided for *e*- transactions.

5. PERFORMANCE ANALYSIS OF CRYPTOGRAPHIC ALGORITHMS

The proposed approach provides a unified approach to secure file transfer by employing a performance analysis of the various cryptographic algorithms. For the study of various algorithms, one needs to perform a parametric study. The two main parameters for any available algorithm are the time complexity and the space complexity. Our study mainly deals with the time complexity since it's difficult to simulate the space complexity considering the cost and amount of work required.

We define time complexity of a problem as the number of steps that it takes to solve an instance of the problem as a function of the size of the input (usually measured in bits), using the most efficient algorithm. To understand this intuitively, consider the example of an instance that is n bits long that can be solved in n^2 steps. In this example we say the problem has a time complexity of n^2. Of course, the exact number of steps will depend on exactly what machine orlanguage is being used. To avoid that problem, the Big O notation is generally used. If a problem has time complexity O (n^2) on one typical computer, then it will also have complexity O (n^2) on most other computers, so this notation allows us to generalize away from the details of a particular computer.

The space complexity of a problem is a related concept that measures the amount of space, or memory required by the algorithm. An informal analogy would be the amount of scratch paper needed while working out a problem with pen and paper. Space complexity is also measured with Big O notation. The following hashing algorithms were taken into account viz., MD2, MD5, SHA1 and SHA256.

First, a fixed size file was given as input. This is the choice obtained from the user. Then, on selection of a particular algorithm, we compute the algorithmic efficiency based on time complexity. The key values used are the time taken for generation of the hash, length of the hash and the number of computational rounds of the algorithm. The greater the length of the hash generated with lesser computation (lesser number of rounds), the better is the algorithm, and vice versa. Now taking these values for a given size of input, we can easily say which one works the best for our application. So there is application-independence for this particular analysis.

The text file is to be taken as input and other formats are not allowed. This comparison of calculated values of efficiency for the various algorithms is based on parameters like the hash length and the number of computations required for hash calculation. The time taken for each algorithm MD2, MD5, SHA and SHA256 are as follows: 501, 204, 141 and 157. The Hash length for each of the above algorithms are 16,15,20 and 3 respectively.

For our *e*-business application, SHA1 is found to be the most efficient (77%) and hence have implemented the same for digest creation.

6. CONCLUSION

In this paper, unified approach was proposed for securing electronic data transfer over e-business transactions. As an analysis, we evaluated the performance of various hashing algorithms available and chose the one most suited for our application. This unified approach for security implementation has been illustrated in our system using an e-business application. In future it can be extended to several other applications like medical transcription, legal documents and others. We could have multiple receivers transacting simultaneously with a sender. The sender, in this case, must be able to authenticate each of the receivers not only at the application level but also at other levels.

The proposed work has been implemented for HTTP protocol based transfers alone. As a future work, secure transfers over protocols like FTP and SMTP can also be implemented, to provide extended usability to the user.

7. REFERENCES

1. Saad M. Darwish, Ahmed M. Hassan, 2012, A model to authenticate requests for online banking transactions, Alexandria Engineering Journal, Volume 51, Issue 3, Pages 185-191.
2. Hakim Fourar-Laidi, A smart card based framework for securing e- business transactions in distributed systems, Journal of King Saud University - Computer and Information Sciences, Volume 25, Issue 1, January 2013, Pages 1-5
3. Said S. Al-Gahtani, Modeling the electronic transactions acceptance using an extended technology acceptance model, Applied Computingand Informatics, Volume 9, Issue 1, January 2011, Pages 47 -77.
4. William Stallings, 2003, "Cryptography and Network Security," McGraw Hill Publisher.
5. Kyle Meadors, 2005, "Secure Electronic Data Interchange Over the Internet", IEEE Computer Society.
6. Piva, Fabio, and Ricardo Dahab,2011, "E-commerce and fair exchange- The problem of item validation",Proceedings of the International Conference on Security and Cryptography (SECRYPT), IEEE, 2011.
7. Haque, A. K. M. Ahasanul and Ismail, Ahmad Zaki and Daraz, Abu Hayat (2009) Issues of E-banking transaction: An empirical investigation on Malaysian customers perception. Journal of Applied Sciences, 9 (10). pp. 1870-1879
8. Nenadic, Aleksandra, Ning Zhang, and Stephen Barton. "FIDES–a middleware e-commerce security solution." Proceedings of the 3rd European Conference on Information Warfare and Security (ECIW 2004), London, UK. 2004.
9. Pradnya B. Rane, Meshram, B, Transaction security for e-commerce applications, International Journal of Electronics and Computer Science Engineering, Vol-1, No.3, pp. 1720-1726.
10. Robinson, Paul, Nick Cook, and Santosh Shrivastava. "Implementing fair non-repudiable interactions with Web services." EDOC Enterprise Computing Conference, 2005 Ninth IEEE International. IEEE, 2005.